Leveraging Entity Types and Properties for Knowledge Graph Exploitation

Alberto Tonon[1]

eXascale Infolab, University of Fribourg (Switzerland)
alberto.tonon@exascale.info

1 Motivation

The last fifty years have witnessed a radical change in the way knowledge is gathered, stored, and represented. Starting from the 70s much effort was put into fitting knowledge into relational tables, and into defining strict schemata describing what information each table contains and how it is modeled. This trend begun to decline at the beginning of the new millennium when "Not only SQL" (NoSQL) databases, exploring other alternatives for modeling, storing, and retrieving information, were designed. The adoption of such alternatives accelerated as a consequence of the needs of big Web 2.0 players, and evolved about ten years ago with the adoption of entity-centric data modeled as a graph: *Knowledge Graphs* (KGs). As the reader can guess, KGs are composed by nodes and labeled edges and, similarly to semantic networks, nodes represent concepts (*entities*) of the domain taken into consideration, while edges represent relations between them.

Knowledge graphs have been embraced by important Web enterprises: Google products are enhanced by the Google Knowledge Graph; [1] Microsoft developed a knowledge graph called Microsoft Knowledge Graph, which is used to empower Microsoft's products such as Excel and Cortana;[2] Facebook is building its Entity Graph by collecting facts from Wikipedia and from its users, who are asked to provide pieces of data that will probably be incorporated into such a graph;[3] LinkedIn is organizing information about its users in a knowledge graph;[4] Yandex, Baidu and Yahoo! are also investing in such technologies.[5]

Exploiting knowledge graphs is often not trivial: even the task of retrieving an entity without knowing its unique identifier can be challenging. Additional tasks such as detecting entities in unstructured data are also very difficult due to ambiguity and duplicate information. Moreover, Knowledge Graphs can be very

[1] `https://www.google.com/intl/es419/insidesearch/features/search/knowledge.html`

[2] `https://developer.microsoft.com/en-us/graph`

[3] `https://www.technologyreview.com/s/511591/facebook-nudges-users-to-catalog-the-real-world/`

[4] `https://engineering.linkedin.com/blog/2016/10/building-the-linkedin-knowledge-graph`

[5] `https://goo.gl/o9OZ2S` (from `http://ir.baidu.com`), `https://goo.gl/crjZRA` (from `http://searchengineland.com`).

large and require specific algorithms to decide which information about their entities should be provided to their users, and efficient data structure allowing for fast retrieval of information.

Maintaining a knowledge graph is demanding, too. To automatically add data into a knowledge graph one has to decide if the new information is true or not, and appropriate nodes and edges must be created in order to encode the new content. In addition, due to their "data-first, schema later" policy, keeping data and schema aligned in a knowledge graph is, in the general case, not straightforward.

The research work presented for evaluation covers both types of issues connected to KGs. On the one hand, we designed enabler methods allowing users to effectively exploit the information contained in a KG and we worked on a real word scenario showing how the knowledge we acquired can be used. On the other hand, we defined and studied procedures that can be used for maintaining KGs and for fairly evaluating enabler methods.

2 Scientific Contributions

During our research work we two tasks enabling people to leverage Knowledge Graphs in other applications, and one task related to knowledge graph maintenance. Additionally, we developed and evaluated an application of knowledge graphs to event detection and, finally, we studied the common denominator of all our contributions, that is, the use of test collections built by using crowdsourcing.

In the following we give a succinct overview of the tasks we studied, and of the contributions we made. We also mention peer reviewed articles published along our research work.

Entity Retrieval The first enabler method we consider is related to *Ad-hoc Object Retrieval* (AOR, for short) which sums up to building a search engine for entities contained in a given knowledge graph allowing users to retrieve entities by keyword queries. AOR is motivated by the fact that often users of knowledge graphs do not know the identifiers of the entities they are searching but rather know some other information about them that can be simply expressed by using keywords (e.g., "the First Lady").

The solution that we propose makes use both of Information Retrieval techniques and of structured repositories allowing us to explore knowledge graphs. Specifically, we index the entities composing the knowledge graph by using inverted indexes and then adopt standard IR ranking functions to obtain an initial ranked list of answers (entities) to the users' keyword queries. We then use the structured repository to explore the surroundings of each retrieved entity to look for other entities related to the user query, or to collect further evidence on the relevance of entities already retrieved.

A large part of what we present in that context was also published at the SIGIR conference in 2012 [5].

Entity Type Ranking After an entity is retrieved, for example by using the method for AOR described previously, it is often summarized for the end-user. Summarizing an entity means selecting which information is better to show to end-users to describe the entity. This task is motivated by the fact that in current knowledge graphs entities are sometimes associated with too much information, which human users cannot process quickly: some nodes in a knowledge graph can have hundreds or thousands of incoming or outgoing edges describing the entities they represent. Among all the information that can be attached to entities, information on their types is of vital importance as it defines what the entities are; for example, in DBpedia the entity Georges Python[6] is defined to be a person, a politician, and several other things, including human, deputy, and thing. Nevertheless, as the reader might have notices, entities are often associated with several types (Barack Obama has 119 types in DBpedia), and selecting the one type to show to the user can be challenging as it can depend on the context in which the entity appears, on the user who looked for it, etc.

For this reason we tackle a task connected to entity summarization that we call *entity Type Ranking* (TRank, for short). TRank consists in ranking the types of a given entity appearing in a given textual context. In this thesis, we report on how different methods for type ranking perform by studying how they rank entity types appearing in news articles. The approaches we analyze take into consideration several factors including the textual content in which entities are mentioned, relations between types in the knowledge graph (e.g., a politician is also a person), and between entities. We also describe a pipeline for detecting entities and ranking their types that runs on a cluster of computers, and we test its scalability by processing a large dataset composed of more than one million webpages.

Our contributions in this area were published at the International Semantic Web Conference [2],[7] and in the special issue on knowledge graphs of the Journal of Web Semantics [3].

Exploiting Knowledge Graphs for Detecting Events One of the goals of this thesis is to show how knowledge graphs, and semantic technologies in general, can be applied in order to solve various tasks. To do so, we show how we exploited knowledge graphs to solve the task of detecting newsworthy events in Twitter. The system we propose, `ArmaTweet`, takes as input semantic queries describing precisely what kind of events the user wants to detect (e.g., “deaths of politicians”), and constantly monitors Twitter to identify relevant tweets that are then shown to the user together with a semantic summary of the event they describe. Relevant tweets are identified by using Entity Linking techniques to link entity mentions to DBpedia nodes, and state-of-the-art Natural Language Processing tools to extract relations between entities.

`ArmaTweet` is the result of a collaboration between Oxford University, Armasuisse, the R&D agency of the Swiss Armed Forced, and the University of

[6] `http://fr.dbpedia.org/page/Georges_Python`

[7] our article was a best paper award candidate

Fribourg. The results we present in this context were also published in the proceedings of the Industrial Track of 14th edition of the Extended Semantic Web Conference (ESWC) in 2017 [4].

Schema Adherence We now change our point of view and focus on maintenance operations. As we mentioned previously, type information is essential in knowledge graphs but, while earlier we focused on nodes, we now focus on edges. We call each edge in a knowledge graph a *property* of the entity it departs from. It is possible to formally describe each property by defining which type of entities it can connect. In DBpedia, for example, the property "fastest driver" has to connect an entity of type "Grand Prix" to an entity of type "Person". In this case we say that "Grand Prix" is the *domain* of the property, and that "Person" is its *range*. Having data that complies to its schema, in our case edges connecting entities of specific types, allows users to better exploit the information contained in the knowledge graph and is beneficial to many tasks.

During our research work, we tackle the problem of detecting misused and unspecified properties, that is, properties that are either used to describe entities of the wrong type, or lack a domain or range specification. Our contribution in this context consists in a method that leverages the statistical notion of entropy to detect property misuse, and to suggest modifications to the knowledge graph to make it more compliant with its schema.

Results on the effectiveness of our method were presented in 2015 at the Linked Data On the Web workshop (LDOW) [1].

Evaluating Entity Retrieval Systems The last contribution we want to mention is not directly related to techniques that exploit knowledge graphs, but is rather connected to how such techniques are evaluated. During our research we focused on how to use crowdsourcing to fairly and continuously evaluate information retrieval systems. We briefly describe our contribution by using as example the task of Ad-hoc Object Retrieval that we introduced previously. Typically, academic approaches for AOR are evaluated by using test collections composed of a knowledge graph, a set of keyword queries, and a set of relevance judgments specifying if a given entry of the knowledge graph is relevant to a certain query or not. Nevertheless, recall that knowledge graphs are typically very large, so, labeling each possible combination of entities and queries is unfeasible. For this reason, current evaluation initiatives adopt a technique called *pooling* to select which documents to evaluate. In practice, all systems participating to an evaluation initiative are run on the given queries and create ranked list of entities retrieved from the input graph. The top-n documents retrieved by each system for each query will compose a pool of documents that human annotators will judge. Obviously, other systems willing to compare their results against those achieved by systems participating in the evaluation initiative might not have their top-n documents judged, and are thus penalized compared to the original participants. In the thesis submitted for evaluation, we study this phenomenon and we discuss how crowdsourcing can be used in order to mitigate it. The results we present were also published in the Information Retrieval Journal [6].

3 Conclusion and Future Work

Despite much effort has been put into research on methods for maintaining and exploiting knowledge graphs, we believe that still much work can be done on the subject. In particular, we noticed that smaller companies are also starting leveraging such a technology in order to interlink data coming from different silos to improve their products, or for business intelligence purposes. Although all the techniques we briefly introduce in this document work reasonably well on fairly large public dataset, it is not clear how they can perform on smaller graph containing knowledge specific to a company.

In addition, we acknowledge the advent of actionable knowledge graphs, that is, KGs including information on how to perform actions on their entities (e.g. watch a movie, request a service, etc.) To the best of our knowledge, effective maintenance methods allowing the automatic population of KGs with such actions, and enabler methods that effectively exploit such new feature are missing. This opens the door to a whole new branch of research on Knowledge Graphs.

References

1. A. Tonon, M. Catasta, G. Demartini, and P. Cudré-Mauroux. Fixing the domain and range of properties in linked data by context disambiguation. In *Proceedings of the Workshop on Linked Data on the Web, LDOW 2015, co-located with the 24th International World Wide Web Conference (WWW 2015), Florence, Italy, May 19th, 2015.*, 2015.
2. A. Tonon, M. Catasta, G. Demartini, P. Cudré-Mauroux, and K. Aberer. Trank: Ranking entity types using the web of data. In *International Semantic Web Conference (1)*, volume 8218 of *Lecture Notes in Computer Science*, pages 640–656. Springer, 2013.
3. A. Tonon, M. Catasta, R. Prokofyev, G. Demartini, K. Aberer, and P. Cudré-Mauroux. Contextualized ranking of entity types based on knowledge graphs. *J. Web Sem.*, 37-38:170–183, 2016.
4. A. Tonon, P. Cudré-Mauroux, A. Blarer, V. Lenders, and B. Motik. ArmaTweet: Detecting events by semantic tweet analysis. In *Proceedings of the Extended Semantic Web Conference (ESWC)*, 2017.
5. A. Tonon, G. Demartini, and P. Cudré-Mauroux. Combining inverted indices and structured search for ad-hoc object retrieval. In *SIGIR*, pages 125–134. ACM, 2012.
6. A. Tonon, G. Demartini, and P. Cudré-Mauroux. Pooling-based continuous evaluation of information retrieval systems. *Inf. Retr. Journal*, 18(5):445–472, 2015.